Université de Jijel
Département d'électronique
ESE16/TP N4
Email: adelmellit2013@gmail.com

FILTRAGE NUMERIQUE
PARTIE 2 FILTRE RII (réponse impulsionnelle infinie)

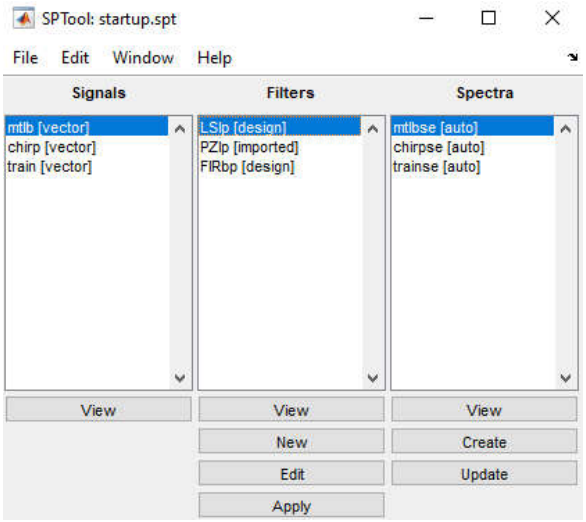
Objectif : C'est la suite de TP 3, qui a pour objectif de réaliser et implémenter un filtre numérique de type RII sur la carte d'évaluation DSK6713.

Matériels : Carte DSK6713, Matlab et CCS 5.5

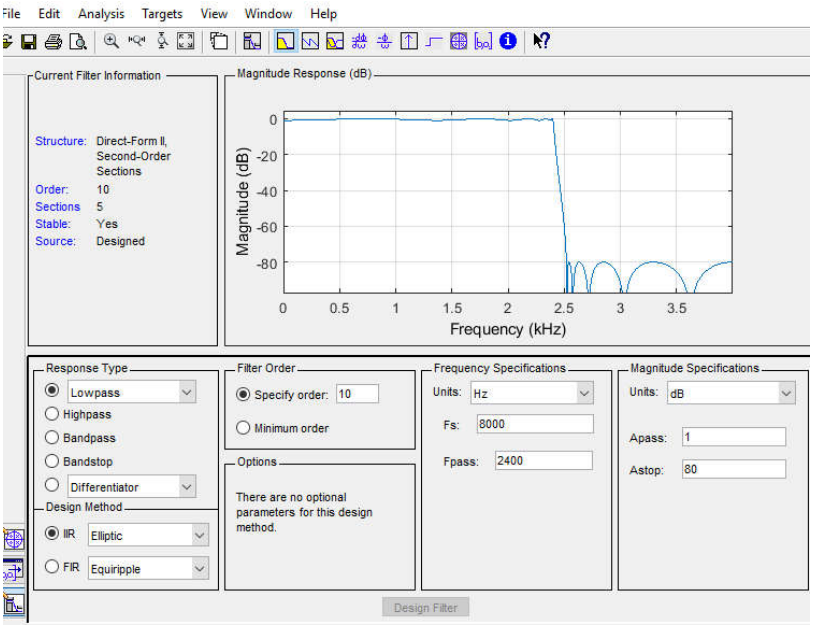
Prérequis : Filtrage numériques sous Matlab, Matlab/Simulink, Langages de programmation C, ou C^{++}

Expérience 1 (Filtre IIR passe-bas)

Dans cette expérience, vous allez concevoir un filtre passe-bas dont la fréquence de coupure est fc = 2400 Hz. Afin de concevoir ce filtre; les coefficients de filtre doivent d'abord être déterminés en utilisant **SPTool** MATLAB



Dans la fenêtre de démarrage; deuxième colonne; sélectionnez nouveau (New). Une fenêtre de conception telle que celle illustrée dans la figure suivante est apparaitre. Dans cette fenêtre, sélectionnez le type de filtre (IIR elliptique), définissez la réponse sur passe-bas et spécifiez l'ordre du filtre à 10. Réglez la fréquence d'échantillonnage sur fs = 8 kHz et la fréquence de coupure fc = 2400 Hz.

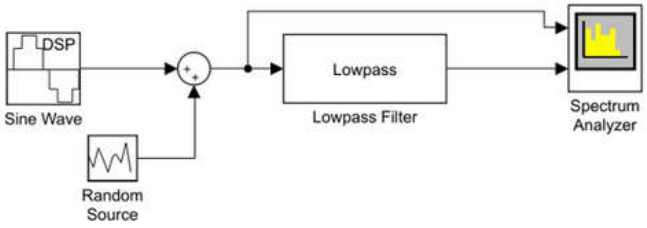


✓ Dans SPTool, sélectionner →Edit→Name, Changer le nom (IIR2400)

✓ Sélectionner: →Export → Export to workspace le fichier IIR2400.

✓ Dans le MATLAB's workspace taper les commandes suivantes:

```
>>[z,p,k] = tf2zp(IIR2400.tf.num, IIR2400.tf.den);
>>sec_ord_sec = zp2sos(z,p,k);
>>sec_ord_sec = round(sec_ord_sec*2^15)
```



La première commande (*sec_ord_sec = zp2sos(z,p,k)*) trouve les racines du numérateur et du dénominateur (zéros et pôles). La deuxième commande (*sec_ord_sec = zp2sos(z,p,k)*) convertit les coefficients à virgule flottante résultants en un format pour l'implémentation en tant que sections de second ordre. La troisième commande (*sec_ord_sec = round(sec_ord_sec*2^15)*) met à l'échelle ces coefficients pour une implémentation à virgule fixe.

```
27585   -10772   27585   32768   -11329   25257
32768   -12180   32768   32768    -9065   31465
32768   -13408   32768   32768   -15948   31492
32768   -11823   32768   32768   -10215   32554
32768   -13762   32768   32768   -15253   32557
```

Ces 30 coefficients représentent les coefficients du numérateur a0, a1 et a2 et les coefficients du dénominateur b0, b1 et b2. Ils représentent six coefficients par étape, avec b0 normalisé à 1 et mis à l'échelle par $2^{15} = 32768$.

Les coefficients utilisant SPTool doivent être contenus dans le fichier IIR2400.cof, répertorié ci-dessous. Ce fichier présente 25 coefficients (au lieu de 30).

Le coefficient b0 étant toujours normalisé à 1, il n'est pas utilisé dans le programme.

```
//IIR200.cof  IIR low pass coefficient file, with cut off at 2400Hz

define stages 5                                          //number of 2nd-order stages
int a[stages][3]= {                                      //numerator coefficients
{27585, -10772, 27585},                                  //a10, a11, a12 for 1st stage
{32768,-12180, 32768},                                    //a20, a21, a22 for 2nd stage
{32768, -13408, 32768},                                   //a30, a31, a32 for 3rd stage
{32768, -11823, 32768},                                   //a40, a41, a42 for 4th stage
{32768, -13762, 32768}
 };
int b[stages][2]= {                                       //denominator coefficients
{-11329     , 25257},                                    //b11, b12 for 1st stage
 {-9065, 31465},                                          //b21, b22 for 2nd stage
{-15948     , 31492},                                     //b31, b32 for 3rd stage
{-10215    ,   32554},                                   //b41, b42 for 4th stage
{-15253,   32557}                                        //b51, b52 for 5th stage
 };
```

Pour implémenter le filtre IIR sur le kit DSK6713, ouvrez le projet existant sur lequel vous avez travaillé la dernière fois et ajoutez-y le code C suivant.

```
//IIR.c IIR filter using cascaded Direct Form II
//Coefficients a's and b's correspond to b's and a's from MATLAB
#include "DSK6713_AIC23.h"                                  //codec-DSK support file
Uint32 fs=DSK6713_AIC23_FREQ_8KHZ;                          //set sampling rate
#include "lp2400.cof"                                       //low pass @ 2400 Hz coefficient file
short dly[stages][2] = {0};                                 //delay samples per stage
interrupt void c_int11()                                    //ISR
{
short i, input;
int un, yn;
input = input_sample();                                     //input to 1st stage
for (i = 0; i < stages; i++)                                //repeat for each stage
{
un=input-((b[i][0]*dly[i][0])>>15) - ((b[i][1]*dly[i][1])>>15);
yn=((a[i][0]*un)>>15)+((a[i][1]*dly[i][0])>>15)+((a[i][2]*dly[i][1])>>15);
dly[i][1] = dly[i][0];                                      //update delays
dly[i][0] = un;                                             //update delays
input = yn;                                                 //intermed out->in to next stage
}
output_sample((short)yn);                                   //output final result for time n
return;                                                     //return from ISR
}
void main()
{
comm_intr();                                                //init DSK, codec, McBSP
while(1);                                                   //infinite loop
}
```

- ✓ Après avoir créé le projet, chargez-le sur le kit DSK6713, puis exécutez le programme.
- ✓ Pour tester le filtre conçu, connectez une onde sinusoïdale du générateur de fonctions à la borne LINE IN du kit DSK6713.
- ✓ Réglez l'amplitude de l'onde de signe sur 1 Vcc et faites varier la fréquence de l'onde sinusoïdale de 50 Hz à 3,4 kHz par un pas de 200 Hz.
- ✓ Mesurer l'amplitude de l'onde de sortie à chaque point de fréquence et tabuler vos résultats comme indiqué dans le tableau suivant.

freq	50	400	600	1000	1400	1600	1800	2000	2200	2400	2600	3000	3200	3400
Am														

- ✓ Tracez l'amplitude en fonction de la fréquence et commentez les résultats.

Travail demandé

Répéter le processus, et concevoir un passe-haut avec une fréquence de coupure de 2200 Hz, un filtre passe-bande avec une fréquence centrale de 1750 Hz et un filtre coupe-bande avec un arrêt de bande centré sur une fréquence de 1750 Hz.

---

[1] **Référence :**

1. https://www.mathworks.com/matlabcentral/fileexchange/19770-simulink-labs-based-on-dsp-first-labs
2. https://eng.najah.edu/media/filer_public/00/c2/00c2c468-b7da-4c0d-9619-eeb1a77a5b07/
3. https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/s/89952_93049v00_Simulink_Real-Time_Whitepaper.pdf
4. Chassaing, R. (2004). Digital Signal Processing and Applications with the C6713 and C6416 DSK (Vol. 16). John Wiley & Sons.
5. Kumar, B. P. (2016). Digital signal processing laboratory. CRC press.
6. https://www.youtube.com/watch?v=9DGjAKEB0eU // TMS320C6713 DSK Tutorial 7A - FIR and IIR filter design using MATLAB & SIMULINK